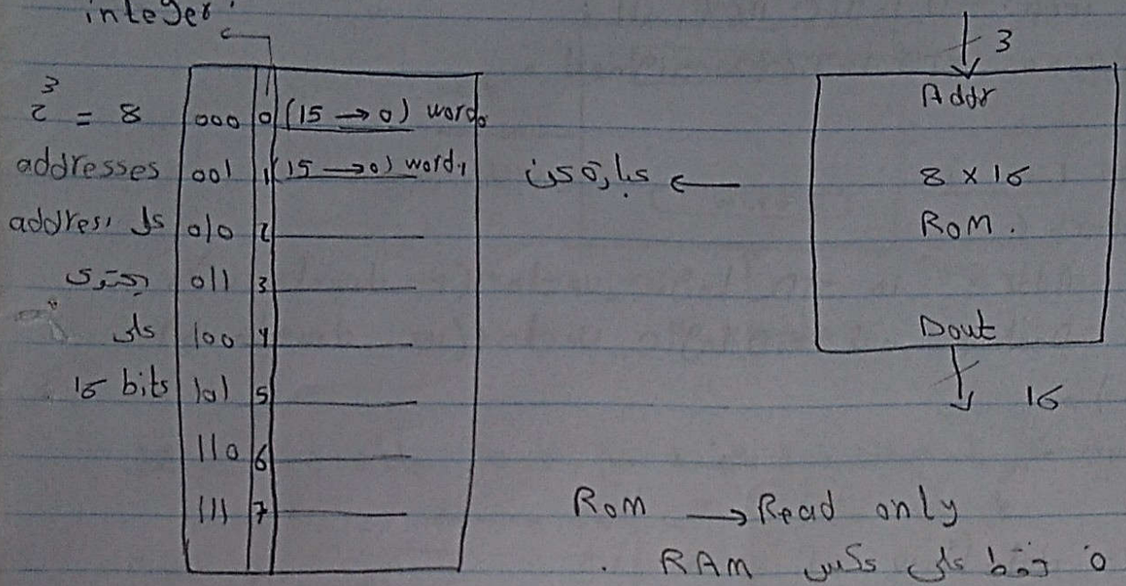
VHDL examples

① Rom 8 x 16 ← 8 addresses, each address contains word of 16 bits

integer;



Rom → Read only

لكن تزاد فقط على عكس RAM.

تزاد من address معين

لكن جلب أي قيمة من الـ Rom في الـ Dout.

* memory(0) = word0
  memory(1) = word1
  ⋮
  memory(7) = word7

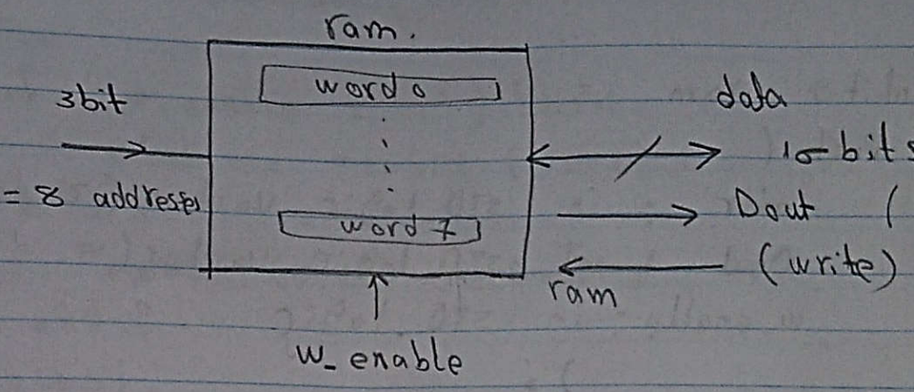
memory(Temp) where Temp 0 → 7.

(2)

```
Library ieee;
use ieee.STD_Logic_1164.all;
use ieee.STD_Logic_unsigned;

entity rom is
    Port (
        Addr : in STD_Logic_vector(2 downto 0);
        Dout : out STD_logic_vector(15 downto 0)
    );
end rom;

architecture dataflow of rom is
  signal Temp : integer range 0 to 7;
  Type rom_array is array (0 to 7) of STD_logic_vector (15 downto 0);
    constant memory : rom_array := (
                        X"800A",
                        X"0459",
                        X"A412",
                        X"059B",
                        X"CD0F",
                        X"F103",
                        X"6817",
                        X"0013");
  Begin.
  Temp <= to_integer(adder);
    Dout <= memory(temp);
end data flow.
```

RAM

نفس طريقة Rom ولكن هاعل read & write.



نفس البرنامج السابق بالضبط مع زيادة w_enable، عل check عليه لو كان 1 ← write.

Dout → ram.

read. ← 0

ram → Dout.

لكن خذ بالك في الـ ram سماك اللي جايين address(0) أو address(1)

:

address(7)

Dout وخرجه في

0	5617
1	AB03
2	F032
3	EB70
→ 4	0134
5	D213
6	304A
7	BA30.

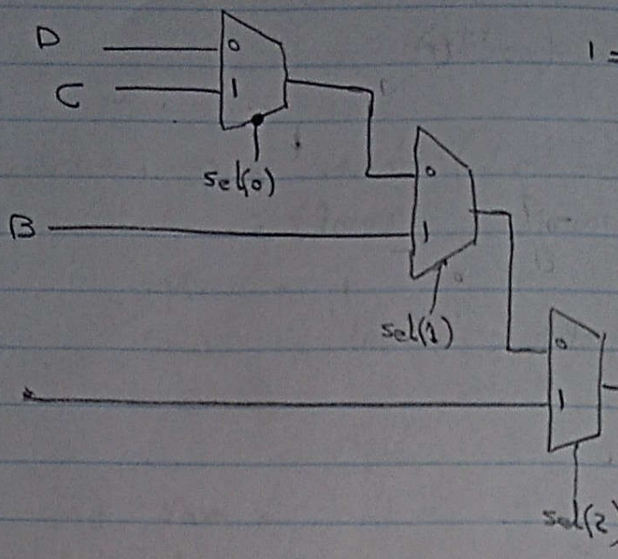
addresses = integer(addr) = 2^3

```
Libray  ieee ;
use     ieee . STD_ LoGiC_ 1164 . all ;
use     ieee . STD_LoGiC_ unsigned ;

entity  ram  is
  Port (
      addr  : in  STD_LoGiC_ vector ( 7 downto 0 ) ;
      Dout  : out STD_ LoGiC_ vector ( 15 downto 0 ) ;
      w_enable : in  STD_LoGiC
              ) ;
  end  ram ;
architecture dataflow of ram is
  signal Temp : integer range 0 to 7 ;
  Type  ram_ array is array (0 to 7) of STD_ LoGiC_ vector.
                                        (15 downto 0) ;

  Constant memory : ram_array ) :=
                    (
                      X" 5617 " ,
                      X" AB03 " ,
                      X" F032 " ,
                      X" EB 70 " ,
                      X" 0134 " ,
                      X" D213 " ,
                      X" 304A " ,
                      X" BA 30 " ) ;
Begin.
```

(5)

```
process ( addr , w_enable , Temp) .
    begin.
        Temp <= to_integer ( addr) .

        if ( w_enable = '0' ) then.
            Dout <= memory ( Temp) ;
        else
            memory (Temp) <= Dout ;
        endif .
    end process
end dataflow ;
```

ex



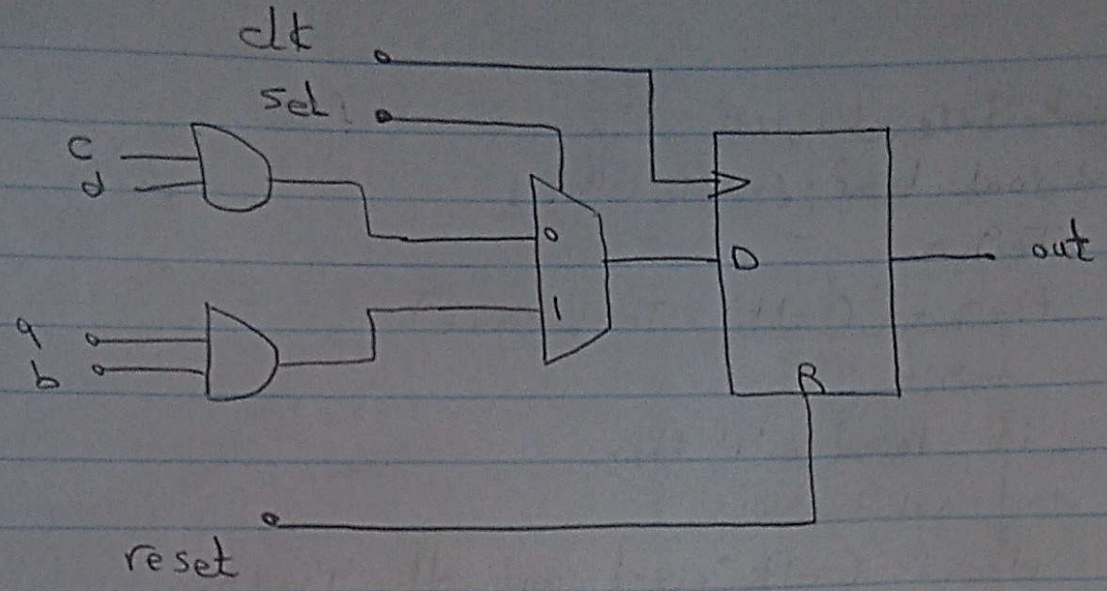
من الرسم باجر دالنظر طالما sel(2) = 1
A هي الخرج دايماً
لكن لو sel(2) = 0 نشوف sel(1)
لو sel(1) = 1 ← B هنددي
اجراً كده كده sel(2) = 0
طيب لو sel(1) = 0 نشوف
sel(0)

```
process ( sel, a , b, c, d)
  begin.
if ( sel(2) ='1') then
     y <= A ;
else if ( sel(1) ='1') then
     y <= B ;
else if ( sel(0) = '1') then
     y <= C ;
 else
     y <= D ;
end if ;
end process ;
```

sel(2)	sel(1)	sel(0)	y
0	0	0	D
0	0	1	C
0	1	0	B
0	1	1	B
1	0	0	A
1	0	1	A
1	1	0	A
1	1	1	A

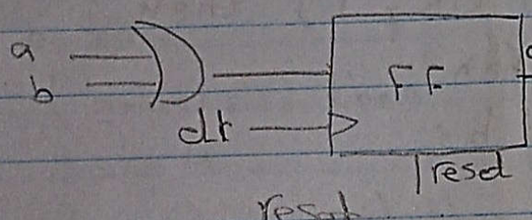
بدأ بـ sel(2) لأنها higher priority.
: نبدأ بـ if
sequential statement
الـ داخل الـ process.

ex




```
Process ( clk, reset , sel)
  begin
  if ( reset = '1') then
      out <= '0' ;
  else if ( clk' event and clk = '1') then.
       if ( sel = '1') then
         out <= a and b;
         else
           out <= c and d ;
         end if ;
    end if ;
    end process ;
```

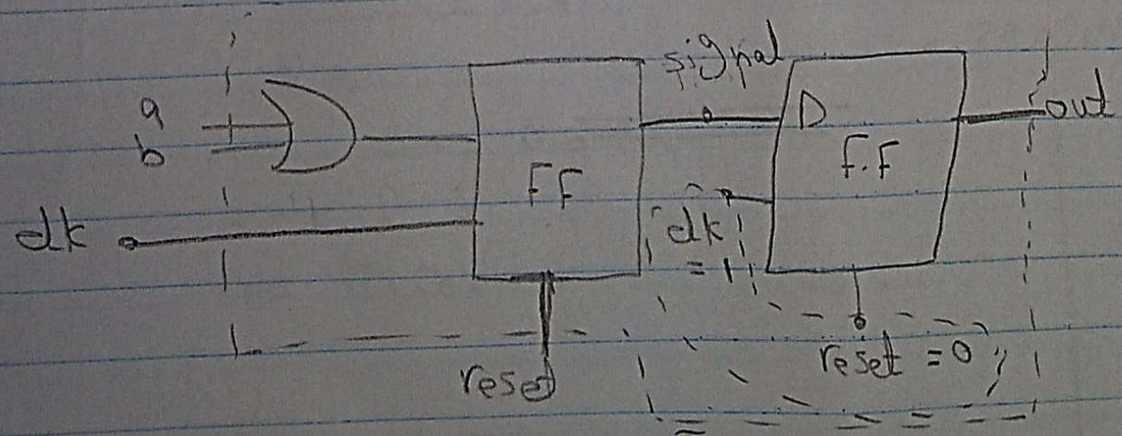
```
architecture var of FF is
  signal temp: std_logic;
  begin.
  Process (clk , Temp , a , b)
   begin.
   if reset = '1' then.
      out <= '0';
   elseif (clk'event and clk= '1') then
     temp <= a or b;
      out <= temp ;
   end Process;
   end var;
```



XXXX

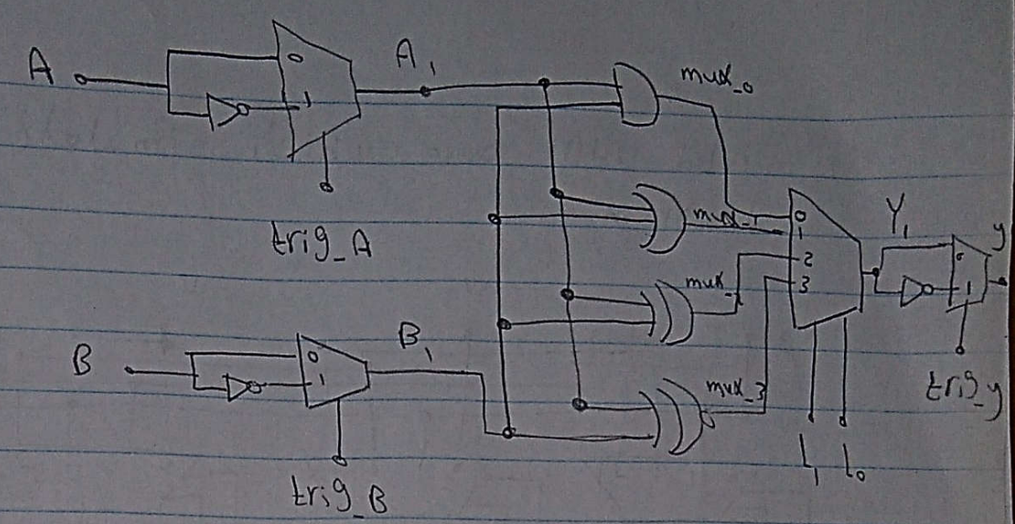
كلك لوجود signal Temp
ومن المدرب عدم ظهور الـ signal
لذلك

هذا البرنامج يعمل من multiple FF.



. Temp الى out يخرج D داخل الـ

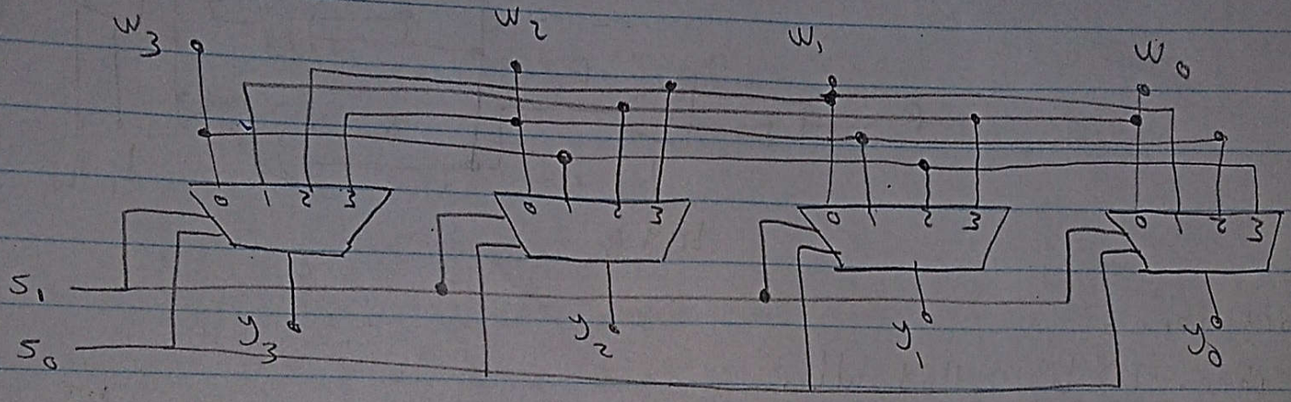
(9)




```
Library ieee;
use ieee.STD_LOGIC_1164.all;
entity mix is
    Port (
        A, B, trig_A, trig_B, trig_y, L1, L0 : in STD_LOGIC;
        y : out STD_LOGIC);
    end mix;
architecture Problem of mix is
    signal A1, B1, mux_0, mux_1, mux_2, mux_3, Y1 : STD_LOGIC;
    signal L : STD_LOGIC_vector (1 downto 0);
Begin   A1 <= NOT A when (trig_A = '1') else
               A;
        B1 <= NOT B when (trig_B = '1') else
               B;
        mux_0 <= A1 and B1;
        mux_1 <= A1 or B1;
        mux_2 <= A1 xor B1;
        mux_3 <= A1 xnor B1;
        L <= L1 & L0;
        with L select
        Y1 <= mux_0 when "00",
              mux_1 when "01",
              mux_2 when "10",
              mux_3 when others;

        y <= NOT Y1 when (trig_y = '1')
             else Y1;
```

(10)

① write VHDL code which simulate the following diagram.



② write VHDL code to make 4 bit up down counter.

③ find errors in the following code.

```
Library ieee ;
use ieee.STD.logic.1164.all;
entity mux is
port ( w1, w2, w3 : in STD_logic ;
       s1, s2     : in STD_logic_vector(1 downto 0);
       f          : out STD_logic ;
)
end mux ;
architecture dataflow of mux is
f <= w1 when s1 = 1 else ;
     w2 when s2 = 1 else ;
     w3 ;
end dataflow ;
```

④ write VHDL code to make 1 bit shift Right and shift left for 4 bit input.